Atlas**Scientific**
Biology • Technology

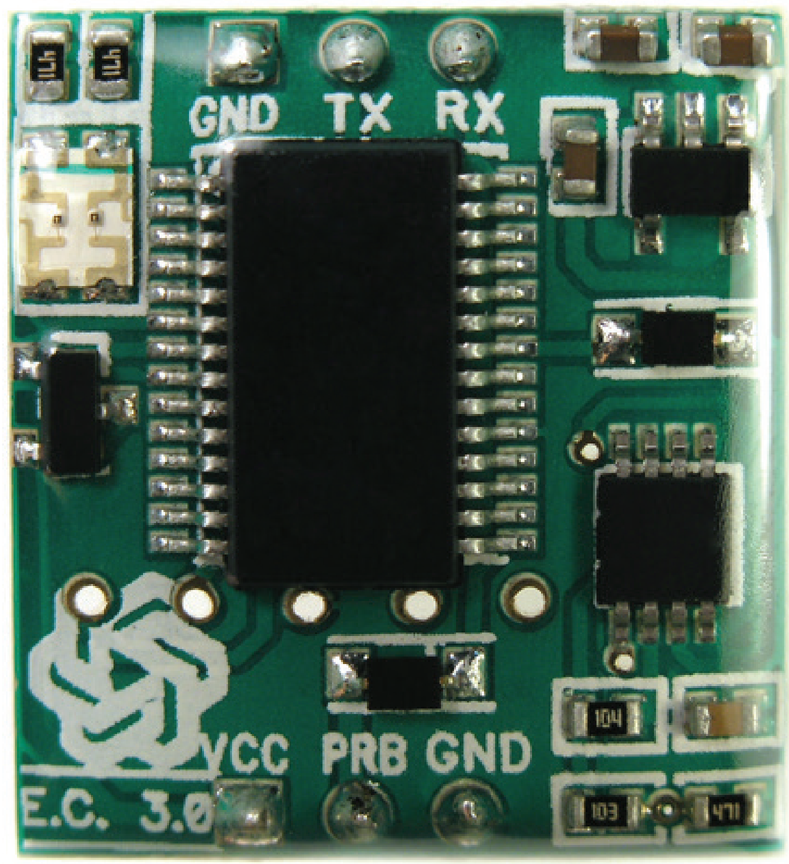
# E.C. Circuit

**V 3.0**

# DISCONTINUED as of 10/21/13

## Features

- E.C. readings +/- 5µs/cm
- Full E.C. capability from 11µs/cm to 92,000µs/cm
- Temperature dependent or temperature independent readings
- Total dissolved solids (TDS) referenced to KCL
- Salinity is derived from the Practical Salinity Scale (PSS-78)
- Data output is a comma separated string: µs/cm, TDS, Salinity
- Single reading or continuous reading modes
- Simple asynchronous serial connectivity **(voltage swing 0-5v)**
- Simple instruction set consisting of only 6 (not including calibration) commands
- Micro footprint circuitry
- Debugging LED's
- 5V operational voltage




4.2 mA in active mode*
3.8 mA in quiescent mode*
*LED's off

## Description

Reading the E.C, TDS and salinity of water is an extremely complicated task, frustrating many embedded systems engineers. Atlas Scientific has taken the complexity out of water quality analysis with respect to E.C/TDS/Salinity with the Atlas Scientific E.C Circuit .

The E.C. Circuit is a highly compact electrical conductivity monitoring system that fits into any breadboard. This design configuration allows the user to accurately monitor E.C./TDS/Salinity without having to add any additional circuitry or components to your design. The E.C. Circuit can take readings from 3 different types of conductivity Sensors (K 0.1/ K 1.0/K 10); giving the E.C. Circuit  a range of 11µs/cm to 92,000µs/cm. Communication with the E.C. Circuit  is done using only 6 simple commands. The E.C Circuit  provides scientific grade readings to any embedded system that has a UART asynchronous serial connection interface.

# Contents

# System Overview

The E.C. Circuit is easy to connect to your micro-controller, requiring only two data lines. 6 simple commands are used to control the E.C. Circuit . It is necessary to connect the E.C. Circuit to a conductivity Sensor with a K constant of 0.1, 1.0 or 10.0 for the E.C. Circuit to work properly. The E.C. Circuit has been designed to operate at 5.0V. Care should be taken to make sure that the power supplied to the E.C. Circuit is as close to 5 volts as possible for accurate readings.

# Sensor type

Three different types of conductivity Sensors can be connected to the E.C. Circuit

Each Sensor will provide high resolution if it is used in the correct type of water.

| Sensor type | Type of water to be analyzed | Sensor range |
|---|---|---|
| K 0.1 | Pure water and drinking water | 11µs/cm to 3,000µs/cm |
| K 1.0 | Fresh water to brackish water | 1,300 µs/cm to 40,000µs/cm |
| K 10 | Salt water | 36,000 µs/cm to 92,000µs/cm |

Using a K 0.1 Sensor in brackish or salt water will return "--"indicating the reading is out of range. Conversely using a K 10 Sensor in drinking water with a conductivity of 100 µs/cm would read "0"; as the Sensor would not be able to detect anything in the water.

# Pin Out

**GND** Return for the DC power supply. GND (& Vcc) must be ripple and noise free for best operation.

**Vcc** Operates at 5.0V

**TX** TX output delivers asynchronous serial data in TTL RS-232 format, except voltages are 0-5v. The output is (up to 17) ASCII digits representing the E.C. in Microsiemens/TDS(in PPM)/Salinity(PSS-78) and ending with a carriage return (ASCII 13).

Example: 50000,32800,32<CR>

50,000μs/cm 32,800ppm 32 salinity

The baud rate is: 38400, 8 bits, no parity, with one stop bit.

If standard voltage level RS232 is desired, connect an RS232 converter such as a MAX232.

# Absolute Maximum Ratings*

| Parameter | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|
| Storage temperature (Micro-E.C. controller) | -40 | | 125 | C° |
| Storage temperature (E.C. Sensor) | -20 | 25 | 125 | C° |
| VCC | 5.0 | 5.0 | 5.0 | V |

***Note:** Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. Exposure to maximum rating conditions for extended periods may affect device reliability

# Device operation

When the E.C. Circuit is connected to a power supply (5.0v) the **green** "power on" indicator LED will be lit. The device will immediately enter standby mode and wait for a command.

There are a total of 8 different commands that can be given to the Micro-E.C. system. (These are operational commands and not calibration commands)

**All commands must be followed by a carriage return <CR>.**
**Commands are not case sensitive.**

## Operational commands: Quick Reference

| Command | Function | Output |
|---|---|---|
| L1 | Enables debugging LEDs | Enabled (default state) |
| L0 | Disables debugging LEDs | Disabled |
| R | Returns a single E.C./TDS/salinity reading. Where the temperature is set to 23 ° Celsius. | 50000,32800,32 |
| TT.T,[C] | Returns a temperature compensated conductivity reading. Adding the optional "C" (25.0,c<CR>) Will return continues readings Q 1000MS | 50000,32800,32 |
| C | Returns continues readings Q 1000ms at the temperature previously used (if temp temperature was not set 23° C is used) | 50000,32800,32 |
| E | Stops all readings. Enter standby/quiescent mode | N/A |
| X | Instructs the E.C. Circuit to do a factory reset | N/A |
| I | Returns device info | N/A |

## Calibration Command: Quick Reference

| Command | Function | Output |
|---|---|---|
| P,[1,2,3] | Sets the Sensor type being used by the circuit | k0.1, k1.0, k10.0 |
| Z0 | Dry Sensor calibration | "Dry Cal" |
| | **K0.1 Sensor calibration** | |
| Z2 | Calibrate for 220 µs/cm | "220 µs/cm cal" |
| Z30 | Calibrate for 3,000 µs/cm | "3,000 µs/cm cal" |
| | **K1.0 Sensor calibration** | |
| Z10 | Calibrate for 10,500 µs/cm | "10,500 µs/cm cal" |
| Z40 | Calibrate for 40,000 µs/cm | "40,000 µs/cm cal" |
| | **K10.0 Sensor calibration** | |
| Z62 | Calibrate for 62,000 µs/cm | "62,000 µs/cm cal" |
| Z90 | Calibrate for 90,000 µs/cm | "90,000 µs/cm cal" |

## Command Definitions

**L1** This will enable both debugging LED's.

The E.C. Circuit has two LED's
**Green LED** .............................Power indicator
**Red LED** .................................Instruction received/E.C. transmit

By default, the LED's are enabled.
These LED's are designed to help the user determine that the E.C. Circuit is operating properly.

Changes to this setting are written to EEPROM memory and therefore will be retained even if the power is cut.

**Keeping the LEDs on will consume an additional 30 mA**

**Full proper syntax: l1<cr> or L1<CR>**

**L0** This will disable both debugging LED's.

Changes to this setting are written to EEPROM memory and therefore will be retained even if the power is cut.

**Full proper syntax: l0<cr> or L0<CR>**

**R** Instructs the E.C. Circuit to return a single E.C. reading.

Taking a single reading is not considered scientifically accurate. The E.C. Circuits readings will become accurate after ~ 15-25 continues readings. A single reading will only give you an estimate of the conductivity.

*This instruction takes 1000 milliseconds to complete

When using the "R" command the temperature is defaulted to 23° C if no temperature data has been entered. If temperature data has been entered then the last temperature entered is used.

**Full proper syntax: r<cr> or R<CR>**

The E.C. Circuit will respond: EC,TDS,SAL<CR>

**Where:**
**EC is the electrical conductivity in μs/cm**
**TDS is Total Dissolved Solids (referenced to Kcl)**
**SAL is Salinity (Practical Salinity scale 1978) expressed as a whole number only.**

If the Sensor responds with "--" the E.C, TDS or salinity is out of range.

Such as:
70000,43000,--

This is because the PSS (practical salinity scale) only goes to 42; which is reached at a conductivity of 62,290 μs/cm

"--"can also be seen when using a Sensor that is not designed for the type of water it is immersed in. Using a K.01 Sensor in saltwater would produce that type of reading.

**TT.TT[,C]** Instructs the E.C. Circuit to return a single temperature compensated E.C. reading.

Inputting a temperature value to the E.C. Circuit will change the circuits default temperature. All new readings will be based off of that new temperature until system reboot.

A temperature can be entered one time or each time a new reading is taken. Placing a ",c" after the temperature instructs the E.C. Circuit to take continues readings after the temperature has been entered

**Temperature is always in Celsius**

**Full proper syntax: 17.8<CR> or 17.8,C<CR>**
**(Where 17.8 is a representation of any Celsius temperature you wish to enter)**

The E.C. Circuit will respond: EC,TDS,SAL<CR>

**Where:**
**EC is the electrical conductivity in µs/cm**
**TDS is Total Dissolved Solids (referenced to Kcl)**
**SAL is Salinity (Practical Salinity scale 1978) expressed as a whole number only.**

**If 17.8,C<CR> was entered the readings would be transmitted from the E.C. Circuit every 1000ms**

If the Sensor responds with "--" the E.C, TDS or salinity is out of range.
Such as: 65000,35100,--

This is because the PSS (practical salinity scale) only goes to 42; which is reached at a conductivity of 62,290 µs/cm

"--"can also be seen when using a Sensor that is not designed for the type of water it is immersed in. Using a K.01 Sensor in saltwater would produce that type of reading

**C** Instructs the E.C. Circuit to take continues reading every 1000ms.

The E.C. Circuits reading will become accurate after ~ 15-25 continues readings.

**Full proper syntax: c<cr> or C<CR>**

The E.C. Circuit will respond: EC,TDS,SAL<CR>

**Where:**
**EC is the electrical conductivity in µs/cm**
**TDS is Total Dissolved Solids (referenced to Kcl)**
**SAL is Salinity (Practical Salinity scale 1978) expressed as a whole number only.**

**E** Instructs the E.C. Circuit to End continuous mode and enter standby/quiescent mode

Delivering this instruction when not in continuous mode will have no effect on the E.C. Circuit

**Full proper syntax: e<cr> or E<CR>**

The E.C. Circuit will respond by ceasing data transmission.
There is no ASCII response to this instruction

**X** Instructs the E.C. Circuit to do a factory reset

Delivering this instruction will clear all calibration data and temperature setting.
The E.C. Circuit then will enter standby mode

**Full proper syntax: x<cr> or X<CR>**

The E.C. Circuit will respond: **Factory reset<CR>**

**I** Returns device info

Delivering this instruction will instruct the E.C. Circuit to transmit it device info.

**Full proper syntax: i<cr> or I<CR>**

The E.C. Circuit will respond: **E,V3.0,4/11<CR>**

**Where:**
**E =E.C. Circuit**
**V3.0= Firmware version**
**4/12= Date firmware was written**

# Calibration Instructions

In order to provide the engineer with the greatest possible accuracy E.C. Circuits now ship uncalibrated. This is because micro voltage changes in your circuit design cannot be compensated for at the factory.

## The E.C. Circuit must be calibrated in the following order:

- **Set Sensor type**
- **Calibrate for a dry Sensor**
- **Calibrate for high side µs/cm reading**
- **Calibrate for low side µs/cm reading**

## All calibration data is stored in the EEPROM memory and will not be lost during power down.

**Step 1: Set Sensor type**

Connect the E.C. Sensor to your E.C. Circuit through the BNC connector. The E.C. Circuit needs to know what type of Sensor it is connected to; there are 3 possible Sensor types to choose from.

K0.1
K1.0
K10.0

The E.C. Circuit will permanently change its configuration settings to the Sensor type that has been specified. Anytime a new Sensor type is connected to the Circuit it must be configured to read from that Sensor.

The command to set the Sensor type is:
"P,1" for a K0.1 E.C Sensor
"P,2" for a K1.0 E.C Sensor
"P,3" for a K10.0 E.C Sensor

**P,[1] [2] [3]** Informs the E.C. Circuit what type of Sensor it is connected to.

The E.C. Circuit is defaulted to Sensor type#2

## Sensor command table

| P,[1][2][3] command | Sensor type | E.C.-Circuit response |
|---|---|---|
| P,1<cr> | K 0.1 | K0.1<cr> |
| P,2<cr> | K1.0 | K1.0<cr> |
| P,3<cr> | K10.0 | K10.0<cr> |

## Sensor type table

| Sensor type | Type of water to be analyzed | Sensor range |
|---|---|---|
| K 0.1 | Pure water and drinking water | 11µs/cm to 3,000µs/cm |
| K 1.0 | Fresh water to brackish water | 1,300 µs/cm to 40,000µs/cm |
| K 10 | Salt water | 36,000 µs/cm to 92,000µs/cm |

**Full proper syntax:**

(In this example we will set the E.C. to a K0.1 Sensor type)

**p,1<cr> or P,1<CR>**

The E.C. Circuit will respond: **k0.1<CR>**

**Step 2: Dry Calibration**

Do not put the Sensor in any liquid. You are going to calibrate your E.C. Circuit for a dry condition. This is much like setting the TARE on a scale.

**Z0** Informs the E.C. Circuit to calibrate for a dry Sensor.

**Full proper syntax:** **z0<CR> or Z0<CR>**

The E.C. Circuit will respond: **Dry Cal<CR>**

**Step 3: Calibrate the E.C. Circuit for high side µs/cm readings**

Each E.C. Sensor type is calibrated against 2 different E.C. solutions. One E.C. calibration solution is on the low side of the Sensors E.C. range and the other calibration solution is on the high side.

| Sensor type | Low side calibration solution | High side calibration solution |
|---|---|---|
| K 0.1 | 220µs/cm | 3,000µs/cm |
| K 1.0 | 10,500µs/cm | 40,000µs/cm |
| K 10 | 62,000µs/cm | 90,000µs/cm |


## Calibration is first done on the high end.

K0.1 first calibration: 3,000µs/cm
K1.0 first calibration: 40,000µs/cm
K10.0 first calibration: 90,000µs/cm

- Place the E.C. Sensor in the high side calibration solution.
- Put your E.C. Circuit in continues mode by transmitting the "C" command.
- Let the E.C. Circuit run in continues mode for 3-5 minutes. The readings will start to stabilize after 3-5 minutes.
- Transmit high side calibration command

| Sensor type | High side calibration solution | High side calibration command |
|---|---|---|
| K 0.1 | 3,000µs/cm | Z30<CR> |
| K 1.0 | 40,000µs/cm | Z40<CR> |
| K 10 | 90,000µs/cm | Z90<CR> |

**Example:**
(Here we calibrate an E.C. Circuit connected to a K10 Sensor.)

**Full proper syntax:** **z90<CR> or Z90<CR>**

The E.C. Circuit will respond: **90,000 µs/cm cal**

**Step 4: Calibrate the E.C. Circuit for low side µs/cm readings**

Now that the E.C. Circuit has been calibrated for the high side, now calibrate it to the low side.

The second calibration event is done on the low side. Do not do low side calibration first.

K0.1 Second calibration: 220µs/cm
K1.0 Second calibration: 10,500µs/cm
K10.0 Second calibration: 62,000µs/cm

- Place the E.C. Sensor in the low side calibration solution.
- Put your E.C. Circuit in continues mode by transmitting the "C" command.
- Let the E.C. Circuit run in continues mode for 3-5 minutes. The readings will start to stabilize after 3-5 minutes.
- Transmit low side calibration command

| **Sensor type** | **Low side calibration solution** | **Low side calibration command** |
|---|---|---|
| K 0.1 | 220µs/cm | Z2<CR> |
| K 1.0 | 10,500µs/cm | Z10<CR> |
| K 10 | 62,000µs/cm/cm | Z62<CR> |

**Example:**
(Here we calibrate an E.C. Circuit connected to a K10 Sensor.)

**Full proper syntax:** **z62<CR> or Z62<CR>**
The E.C. Circuit will respond: **62,000 µs/cm cal**

# Calibration is now complete

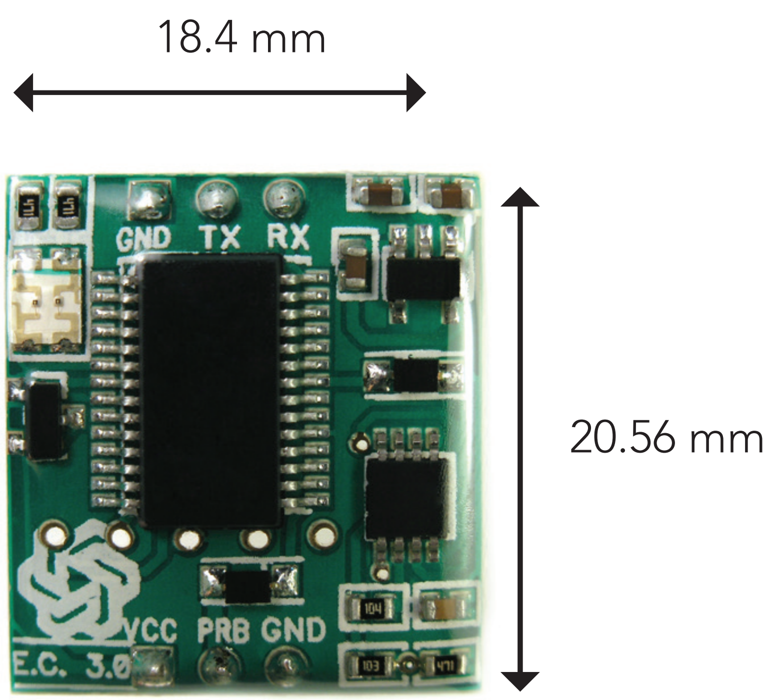
Salinity is derived using: The Practical Salinity Scale 1978 (PSS-78)

$$S = a_0 + a_1 R_T^{1/2} + a_2 R_T + a_3 R_T^{3/2} + a_4 R_T^2 + a_5 R_T^{5/2} + \frac{(T-15)}{1+k(T-15)}\left(b_0 + b_1 R_T^{1/2} + b_2 R_T + b_3 R_T^{3/2} + b_4 R_T^2 + b_5 R_T^{5/2}\right)$$

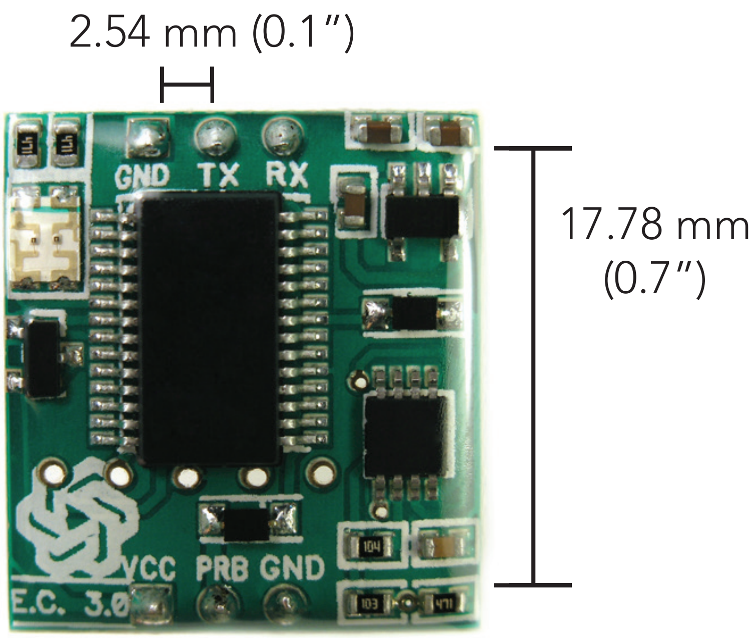
Where: $R = \frac{C(S, T_{68}, P)}{C(35, 15_{68}, 0)}$

Where: $R_T = \frac{R}{r_T R_P}$

Where: $Rp = 1 + \frac{A_1 p + A_2 p^2 + A_3 p^3}{1 + B_1 T + B_2 T^2 + B_3 R + B_4 TR}$

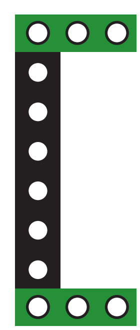


# How to make a Footprint for the Atlas Scientific E.C. Circuit

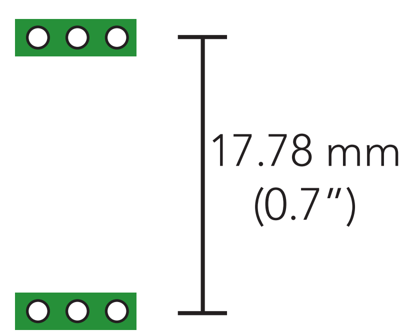


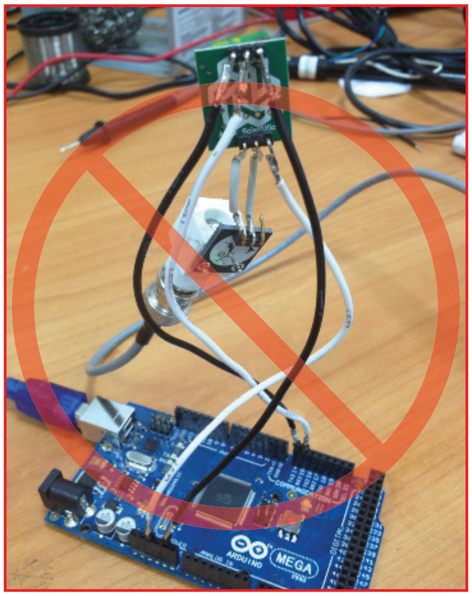

1. In your CAD software place an 8 position header.

2. Place a 3 position header at both top and bottom of the 8 position header as shown.



3. Once this is done you can delete the 8 position header. Make sure that the two 3 position headers are 17.78mm (0.7”) apart from each other.






The Atlas Scientific E.C. Circuit is a piece of senstive equipment. Debugging should be done in a bread board; Not like what is show in this photo.

# Warranty

Atlas Scientific warranty's the E.C. Circuit to be free of defect during the debugging phase of device implementation, or 30 days after receiving the E.C. Circuit (which ever comes first).

# The debugging phase

The debugging phase is defined by Atlas Scientific as the time period when the E.C. Circuit is inserted into a bread board or shield and is connected to a microcontroller according to this wiring diagram. Reference this wiring diagram for a connection to USB debugging device, or if a shield is being used, when it is connected to its carrier board.

If the E.C. Circuit is being debugged in a bread board, the bread board must be devoid of other components. If the E.C. Circuit is being connected to a microcontroller, the microcontroller must be running code that has been designed to drive the E.C. Circuit exclusively and output the E.C. Circuit's data as a serial string.

**It is important for the embedded systems engineer to keep in mind that the following activities will void the E.C. Circuit's warranty:**

- **Soldering any part of the E.C. Circuit**
- **Running any code that does not exclusively drive the E.C. Circuit and output its data in a serial string**
- **Embedding the E.C. Circuit into a custom made device**
- **Removing any potting compound**

# Reasoning behind this warranty

Because Atlas Scientific does not sell consumer electronics; once the device has been embedded into a custom made system, Atlas Scientific cannot possibly warranty the E.C. Circuit against the thousands of possible variables that may cause the E.C. Circuit to no longer function properly.

**Please keep this in mind:**

1. **All Atlas Scientific devices have been designed to be embedded into a custom made system by you, the embedded systems engineer.**
2. **All Atlas Scientific devices have been designed to run indefinitely without failure in the field.**
3. **All Atlas Scientific devices can be soldered into place.**

*Atlas Scientific is simply stating that once the device is being used in your application, Atlas Scientific can no longer take responsibility for the E.C. Circuit continued operation. This is because that would be equivalent to Atlas Scientific taking responsibility over the correct operation of your entire device.